Issue 23 West Los Angeles Atari Users Group Newsletter January 1985 $1.50

NEXT MEETING

The next meeting will be on January 2, 1984, at Paul Revere. The meeting will feature simultaneous exhibitions of arcade style games. The address is:

Room J1
1450 Allenford Ave.
(Go north on 26th St. in Santa Monica--it runs into Allenford)

INDUS RAFFLE

The sale of tickets for the raffle of an INDUS disk drive is not a success. Only 35 tickets have been sold. Rather than wait indefinitely to sell enough tickets to cover the cost of the drive, we will lick our wounds and hold the raffle at the February meeting. The club will take a big loss on this one, unless there is a last minute rush for tickets. Actually, this is a great opportunity to get in on a bargain. Since so few tickets were sold, each ticket is worth much more than the $5 price. So buy a ticket or two right away--it's a bargain!

PRESIDENT'S COLUMN

Let me introduce myself. I am John Scheid, your new group president. I would like to tell you what sort of plans we officers have for this new year. First, on behalf of the entire group, I want to thank our retiring officers, Greg Waldorf, Ken Suzuki and Mitch Paull.

Our new vice president and advertising manager is Arne Per Heurberg. Bob Bruce is staying on as secretary/treasurer. Bill Skurski and Terry Craig will produce our newsletter (Terry does the editing and Bill does the paste-ups and makes it look elegant). Dale Stuck is stuck again with the job of managing our disk library. Dale was our first disk librarian and did a great job then in originating and maintaining the library, and we expect the same from him again. Dick Reaser will stay on as our newsletter librarian.

Our group is in relatively good shape, compared to other Atari user groups across the nation. The limited availability of Atari products means that very few new users are joining user groups.

In spite of this, we are getting a handful of new members each month and membership renewals are fairly steady. Our total membership has decreased somewhat--we are getting fewer of the merely curious or uncommitted new members. As a result, our membership, on the average, is a more serious Atari user.

Your new officers, along with other members who are especially active in group affairs, held a meeting last month to decide on the future course of the club. The consensus was that our group is becoming increasingly more important to its membership in these uncertain times. As the future of Atari is in doubt (especially the dubious prospect for support from Atari of its established products), the group increasingly fulfills the supporting role, serving as a source of information and mutual support. We therefore decided that our mandate is to fill the void left by Atari, Inc.

We can do this by enhancing the services provided by the group. As for information, we intend to seek out more vigorously information about what is going on in Atari and other Atari vendors, and report it all in the newsletter and at the monthly meetings. We will try to step up the number and the quality of hardware and software reviews in our newsletter generated by our own members. Such reviews can be more informative than those appearing in national magazines and other newsletters, especially because the authors can be consulted directly for further information and clarification. We value reviews by our own members very highly, so we strongly urge you to contribute one as soon as you can.

The second service provided by the group is education. I want to raise the level of technical competence of our group. This will be done by providing more technical information in the newsletter, both for neophytes and for the more advanced users among us, and by formal classes in various subjects. You can contribute to raising the technical level of the newsletter by submitting technical articles at any level of sophistication.

So far, we have had two courses on computer programming, one in programming the Atari in assembly language and one on programming in ACTION! I taught both these classes, and they were for more advanced users. I will try to get others to teach courses on other subjects, including several for beginners. These will range from such subjects as basic computer literacy, short courses on spread sheets, word processors and data base systems through advanced programming in several languages. These courses will be offered at very low prices, which will help our group's financial health, too. If you have in mind a subject that you would like us to provide a course for, let me know. And, if there is a subject you feel you can teach and you are willing to donate your talents to such, get in touch with me ASAP!

Another service is that of special interest groups (SIGs). At present, we have one such group, the ATR8000/CPM SIG. This SIG is very successful. They meet at least every four weeks. The meetings are always very lively and well attended. The members help each other on getting familiar with CP/M, learning the various utilities, etc. At last month's meeting, Hans Karge, the SIG's chairman, gave a thorough demonstration of DBASE II. Future meetings are planned, featuring more CP/M utilities, MULTIPLAN, SUPERCALC, WORDSTAR, WORDMASTER, and various programming languages, including COBOL, FORTRAN, PASCAL, C and PL/I. Cooperative projects are planned, such as the development of a copy program that uses the ATR to copy Atari-formatted disks at hitherto unimagined speeds! (Much faster than those devices claiming "turbo" or "warp" speeds, possible because copying would not use Atari's serial bus at all.) A possible future path to expanding this SIG is via the COPOWER 88, a 256K enhancement of the ATR8000, with the 8088 CPU, so that our Ataris will be "IBM PC-compatible".

Following on the success of this SIG, we would like to establish others. How about a hardware SIG? (We might call it SIGHARD.) It could be devoted to learning elementary digital electronics and modifications of the Atari computers and peripherals. It could start out with relatively simple matters, such as adding switches to the computer and peripherals for cold start, recovery from lockup, cartridge backup, writing to the back side of disks. It could also deal with hardware enhancements such as OMNIMON, ramdisks, bank selection, the CHIP, Happy Drive, 80-column boards, and so on. At advanced stages, more elaborate projects such as interfaces to music synthesizers, video equipment and other devices (usually available for other computers, but not for the Atari). Or the construction of complete peripherals.

Or, why not a Programming SIG? This SIG would be concerned with furthering programming in all the available languages, from BASIC through assembly.

Maybe a games SIG. Or a word processing SIG, or home finances, or data bases, or...

If you would like to chair a SIG or help in its formation, or just be a member, let me know. I will help get these SIGs started.

The newsletter constitutes perhaps the most important service of all, certainly the most expensive. It has been somewhat anemic lately, but we will try to improve it with more technical articles, more and better reviews and more general information and gossip. A few other user groups provide excellent material in their newsletters--we will reprint the best of them in our own newsletter. Arne has volunteered to be our advertising manager. He will try to decrease our editorial/advertising content ratio.

Another extremely valuable service provided by our group is the disk library. Dale Stuck is committed to making the library grow once again. He solicits contributions from the members. Dale will also increase the library by the inclusion of programs from other sources. He will also keep us up to date on new acquisitions via the newsletter. You can copy anything in the library, which is located at the Software Supermarket, 11960 Wilshire Blvd.

We have a fast-growing collection

of Atari group newsletters from all across the country. Dick Reaser catalogs and organizes them. Unfortunately, these newsletters are difficult to use, just because there are so many of them--it is hard to find the article you are looking for. A possible solution to this is the preparation of a master index of all the articles in the entire library. This will be a major undertaking, but a very worthwhile one. In the meantime, drop by at Software Supermarket and ask for one or more of the newsletter volumes. You will be amazed at the wealth and variety of information you will find there.

We are working on starting a new bulletin board for our group. It will have a Santa Monica telephone number, both 300 and 1200 baud operation, and will have available for downloading the most popular of the public domain software. Our new vice president, Arne, is in charge of this project.

We have decided to try once again to enlarge the number of local dealers who will offer TRUE discounts to our members. If you have any ideas on how to go about doing this, or would like to help, please call me (phone number below).

Finally, there is the ultimate service, our monthly general meeting. I would like to see each meeting have a theme. Our January theme will be games. We will have several computers set up at the meeting, with two expert gamesman at each station, battling each other at one or more of the most popular arcade-style games. If there is a game you would especially like to see demonstrated or would like to demonstrate a game or be a combatant, call me before the meeting at the phone number below.

Future themes will be among: printers, disk drives, graphics devices and graphics software, data base managers, word processors, home finances, etc.

These are some of our ideas. But ideas are easy to come by. Implementing the ideas is the hard part. It can't all be done by us few officers. If ours is to be an active club, then we must have active members. So get involved, NOW! Just give me a call, or see me after a meeting--I'll find something for you to do.

John Scheid
(213)829-7239 (evenings and weekends)

LAST MEETINGS MINUTES

by
Bob Bruce

We started the meeting at 7:10 with our president Greg Waldorf presiding over his last program. We began with the election of new officers for 1985. Each was introduced and then elected by an unanimous vocal "AYE". So your new officers are as follows -

President-John Scheid
Vice Pres-Arne Heurberg
Sec/Treas-Bob Bruce
Newsletter-Bill Skurski
Assis. Edit-Terry Craig
Librarian-Dale Stuck
Assis. Lib-Joe Spiegel
Historian-Dick Reaser

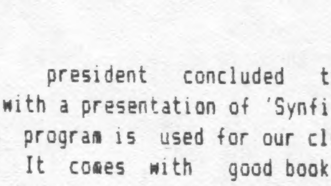
The first of the evenings demonstrations was the second half of the LOGO program by club member Gerry Wick. He pointed out some the unique things that LOGO can be programed to do. There can be three colors on the screen at a time and the turtle shapes can be programed for animation and collision detection which is fun for games. LOGO can be programed for educational quizes with multiple choice questions radomized. Using both words or letters as listings. The joy stick can also be programed for drawing and etc. Thanks again for a great demonstration, Gerry.

It was nice to see Steve Dayan this month (he is our past president). He brought along a new strategy game called "Microleague Baseball". Good animation, good reference books and cards and real team statistics. You can pit 20 real teams against each other or you can build customized teams from real players past and present. It is menu driven and looked like a lot of fun. To order you can call (1-800-PLAYBAL) It will take two weeks for shipping. The cost is $39.95 and other disks of team statistics will become available in the future. Thanks, Steve, for sharing this game with us.

After the break Gerry returned to the computer and and displayed and game he wrote in BASIC for his children. It is called 'Missile Math' and is an arcade type educational game using a joy stick to select answers and shoot missiles at flying space ships. He also booted in 'Music Construction Set' and briefly demonstrated it.

Our president concluded the evening with a presentation of 'Synfile +'. This program is used for our club records. It comes with good books, quick reference cards and a tutorial disk. Greg showed how easy it is to make file templates, retrieve data and make sorts. Thanks, Greg!

After a brief Atari information interchange we ended the evening with a big round of applause to thank our retiring president, Greg Waldorf, for his excellence leadership, his outstanding monthly programs and all of the time he spent arranging for guest speakers at each of our meetings. Thanks - Greg - its been a great year !!!!!!!



BASIC
MOVE

(Retyped by John Scheid)

MOVING BASIC PROGRAMS

by C. Mueller

(reprinted from the Huntsville Atari Users Group newsletter)

When BASIC initializes, it sets its own LOMEM pointer at zero page address $80 and $81, which indicates where a program will load in. When DOS is booted, the operating system LOMEM (at $02E7 and $02E8) is set to point to an address above DOS. Then BASIC uses the O.S. LOMEM to set its own LOMEM pointer to protect DOS from being overwritten by a program. Programs such as BASIC Commander and S.A.M. load in right above DOS, reset the O.S. LOMEM to protect itself, and then initialize BASIC. You can do the same thing yourself even after BASIC has initialized. Just enter the following command:

```
     POKE 744,PEEK(744)+4:NEW
```

The POKE 744 just increased O.S. LOMEM by 1K and the NEW reinitialized BASIC. Check FRE(0) and you'll find that you lost 1024 bytes. But you also gained 1024 bytes of "user memory" - right below the BASIC program area. What this means is that we didn't really need page 6 for user machine language routines, because we can make as much "user memory" as we want. And this memory is persistent - even system reset doesn't alter it.

But how can we create a safe area of RAM from a BASIC program? Well a NEW command obviously destroys a program in memory, so some time ago I wrote a routine to move a BASIC program, pointer and all, anywhere in memory I wanted, while the program was running. The BASIC language itself also moves the program around, with memory management routines which are described in a new book from COMPUTE! called The Atari BASIC Source Book. By using the routines already built into the BASIC language, we can move the program around in memory very easily. WHY would anyone want to move a program around? Well we can have a BASIC program move itself up in memory to create a "hole" in which to place nonrelocatable machine code, or alternate character sets, or screen data which must be protected from being wiped out by the BASIC program. Remember several months ago I said that nonrelocatable code could be carried by a string, which when executed could be made to relocate itself wherever it had to be? Well now we can reserve a place for the code or data. There's another good application of this BASIC program moving technique which involves making more memory. Sometimes we might have a program which fits into memory okay, but as soon as it executes we run out of memory due to dimensioning of variables or opening certain graphics modes. We all know that we lose almost 6K of memory to DOS. Once the program is loaded, if it no longer needs DOS, it could move itself lower in memory, creating an additional 5K to 6K of space. Of course DOS will be gone, but if disk access isn't needed, who cares? And if disk access is needed, the program could be pushed back up in memory and DOS reloaded (yes it can be done).

Here's an example of how to perform this neat trick. Type in the following short program, which will move itself to another location in memory on your command.

```
10 DIM MOV$(11)
20 FOR I=1 TO 11:READ A
30 MOV$(I,I)=CHR$(A):NEXT I
40 DATA 104,104,170,104,168,138
50 DATA 162,128,76,129,168
60 IF PEEK(41007)=122 THEN MOV$(10,10)
= CHR$(124)
70 ADCUR=PEEK(128)+256*PEEK(129)
80 ?"BASIC PROGRAM STARTS AT ADDRESS
";ADCUR
90 ?"THERE ARE ";FRE(0);" BYTES FREE"
100 ?:? "ENTER THE ADDRESS YOU WANT
THIS PROGRAM TO START"
110 INPUT ADNEW
120 OFFSET=ADNEW-ADCUR
130 IF OFFSET<0 THEN MOV$(10,10) =
CHR$(253):IF PEEK(41007)=122 THEN
MOV$(10,10) = CHR$(250)
140 A=USR(ADR(MOV$),ABS(OFFSET))
150 ?:?"THERE ARE NOW ";FRE(0);" BYTES
FREE"
```

The machine language routine MOV$ is set up by the DATA statements in lines 40 and 50. These represent the following short program:

```
     PLA

     PLA

     TAX

     PLA

     TAY

     TXA

     LDX #$80

     JMP $A881
```

All we are doing is setting the CPU registers with values indicating how far to move the program and jumping to the appropriate routine in BASIC. MOV$(10,10) is changed in the BASIC program depending on where the jump to the memory move routine is. For the BASIC cartridge this JMP must be to address $A881 to move the program up in memory and to $A8FD to move the program down in memory. For the built-in BASIC of the 800XL, the JMP must be to $A87C to move it up and to $A8FA to move it down. The BASIC program shown above checks to see which BASIC you have with a PEEK(41007). If this address returns a value of 122 then you have built-in BASIC. If you want to try to use this method of moving BASIC programs remember to change MOV$(10,10) appropriately and pass the number of bytes in memory you want the program to move up or down in the USR call (as in line 140).

CAUTION - The memory management routines in the BASIC cartridge at $A881 and $A8FD are the ones responsible for the famous lock-up. Since lock-up only occurs when a

multiple of 256 bytes is moved, you only have 1/2 of one per cent chance of disaster. SAVE BEFORE RUNNING.
(Retyped by John Scheid)

SCRABLE *GAME*

SCRAMBLE a word game

by Gerry Wick

Here is a little word game that you can play with a friend. Type in a word and the computer scrambles the letters. Then the other player tries to unscramble it. He or she has two guesses. There is even a hint after the first miss. It is a simple program, but lots of fun.

The scramble algorithm is in the subroutine starting at line 1000. Examine it carefully and you can use it to randomize all kinds of things. I have used it to scramble up lists of words in an educational program. Basically what the subroutine gives you is a numerical array with a random distribution of sequential numbers.

There a few other features of the program that I should point out. Notice the REMARK statement in line 10. It serves two purposes. First, by LISTing line 10 and then deleting the 10 and the REM, you can save a program you are working on just by then hitting RETURN. It minimizes typographical errors. Second, you will remember the name you assigned a program by looking at the listing.

In line 300, PEEK(53770) returns a random number which is then POKEd into color registers 710 and 712 flashing the background and the text window through a series of colors. It is a dramatic effect for just a few statements.

Please note that there are three words that should be entered in inverse video. They are "SCRAMBLE" in line 200, "congratulations" in line 420 and "RETURN" in line 430. The listing does not show them that way.

LISTING

```
10 REM SAVE "D:SCRAMBLE.BAS
100 DIM GUES$(20),WORD$(20),
SCRAM$(20),R(21)
110 SCRAM$="":T=0
200 GRAPHICS 0:POSITION 2,8:? "TO PLAY
SCRAMBLE HAVE YOUR FRIEND"
210 ? :? "ENTER A WORD OF 20 LETTERS OR
LESS":? :? "AND THEN YOU TRY TO
UNSCRAMBLE IT"
220 POSITION 2,20:? "ENTER THE WORD
HERE: ":?
230 INPUT WORD$:M=LEN(WORD$)
240 GRAPHICS 2:POKE 712,148
250 GOSUB 1000:REM SCRAMBLE WORD
260 POSITION 10-M/2,2:? #6;SCRAM$
270 ? "TRY TO UNSCRAMBLE THE WORD"
280 IF T=1 THEN ? :? "HERE'S A CLUE:
THE FIRST LETTER IS ";WORD$(1,1)
290 INPUT GUES$
300 IF GUES$=WORD$ THEN FOR N=1 TO
100:P=PEEK(53770):SOUND
0,100-N,10,8:POKE 712,P:POKE 710,P:NEXT
N:GOTO 400
310 FOR N=1 TO 50:SOUND 0,50+N,2,8:NEXT
N:FOR N=1 TO 300:NEXT N:T=T+1:SOUND
0,0,0,0
320 IF T=1 THEN 270
400 SOUND 0,0,0,0:POKE 712,148:POKE
710,148:POSITION 3,4:? #6;"unscrambled
is"
410 POSITION 10-M/2,6:? #6;WORD$
420 IF WORD$=GUES$ THEN POSITION 2,8:?
#6;"congratulations"
430 ? CHR$(125):? "HIT RETURN TO
CONTINUE";:INPUT SCRAM$
440 GOTO 110
1000 REM SCRAMBLE WORD
1010 R(0)=0
1020 FOR N=1 TO M:REM M = LENGTH OF
WORD
1030 VAR=INT(M*RND(0)+1)
1040 R(N)=VAR
1050 FOR P=0 TO N-1
1060 IF R(P)=R(N) THEN 1030
1070 NEXT P:NEXT N
1080 FOR N=1 TO M
1090 SCRAM$(N)=WORD$(R(N),R(N))
1100 NEXT N
1110 IF WORD$=SCRAM$ THEN 1000
1120 RETURN
```

ENCRYPT

A DATA ENCRYPTION ALGORITHM

by Ernie Rice

(Reprinted from the April, 1984 issue of the Jersey Atari Group Newsletter)

This month I am going to provide you with a data encryption algorithm written in BASIC. This routine will provide you with one of the finest encryption routines available. It occupies only 18 lines of BASIC to do both processes and to also check the results. A little explanation is involved before we actually hit the code. The technique used is a proprietary technique being made available to the readers of this publication with the understanding that it will never be used by them in ANY COMMERCIAL application without prior written consent from EHR3, Inc.

Now that that is over and done with, here goes...

The approach is one which allows the user to specify a "combination" to be used for the encryption and decryption of their data. YOU specify the number of digits to be included in the combination (the more digits the safer the data). Up to 255 digits may be used. The combination must be kept safe, as without it your data will never be able to be restored again!

Note also that if you encode a string that contains, say, all A's, they will not all translate to the same character! Thus, an L may represent an A in one place, a W may represent it in another. This makes decoding the data almost impossible (even with another computer at your disposal).

Now, you may say "Why would I ever want to encrypt data?". Good question. Where do you store the Christmas list? What about your own financial records, or personal information? All of these are great candidates for encoding.

The program listed below first prompts for the string to be encoded. You may enter any character string up to 256 bytes in length. Next the program will ask for the number of digits in the computation. Reply with a positive integer value. Next you will be asked for the actual values for each of the digits in the combo; once again positive integers only please.

The program will then display the data in its encrypted form, and verify its translation by encoding it once again and comparing the original and the last strings.

Lines 10000 through 10110 accept the input data from the user. Lines 10400 through 10800 encode the data input using the specified data.

Lines 10900 through 11700 display data, and decode the encrypted string for verification. If an error occurs a mesage "ERROR" will be displayed.

Next month I will show a BASIC program using these routines for simple file use. Until then have a good time encoding...

```
10000 CLR :DIM IN$(256),OUT$(256),
      TRAN$(256),TABLE(128):
      OUT$="":TRAN$=OUT$
10001 ? "STRING OR RETURN";:
      INPUT IN$
10002 IF LEN(IN$)=0 THEN
      FOR A=0 TO 255:
      IN$(A+1,A+1)=CHR$(A):
      NEXT A
10100 INDEX=1:TINDX=1:
      ? "NUMBER OF DIGITS";:
      INPUT DIGITS
10110 FOR F1=1 TO DIGITS:
      ? "DIGIT #";F1;:INPUT X:
      TABLE(F1)=X:NEXT F1
10400 V1=ASC(IN$(INDEX))+INDEX
      +TABLE(TINDX)+DIGITS+
      TINDX+LEN(IN$)
10500 IF V1>255 THEN V1=V1-256:
      GOTO 10500
10600 OUT$(INDEX,INDEX)=CHR$(V1)
10700 TINDX=TINDX+1:
      IF TINDX>DIGITS THEN TINDX=1
10800 INDEX=INDEX+1:
      IF INDEX<=LEN(IN$) THEN 10400
10900 ? OUT$
11000 INDEX=1:TINDX=1
11100 V1=ASC(OUT$(INDEX))-INDEX
      -TABLE(TINDX)-DIGITS
      -TINDX-LEN(IN$)
11200 IF V1<0 THEN V1=V1+256:
      GOTO 11200
11300 TRAN$(INDEX,INDEX)=CHR$(V1)
11400 TINDX=TINDX+1:
      IF TINDX>DIGITS THEN TINDX=1
11500 INDEX=INDEX+1:
      IF INDEX<=LEN(IN$) THEN 11100
11700 IF IN$<>TRAN$ THEN ? "}ERROR"
```

80 COL. BASIC XL

80 COLUMNS AND OSS BASIC XL

By Benjamin Eng

(Reprinted from the Jersey Atari Computer Group Newsletter, April 1984)

The new 16K BASIC XL cartridge from Optimized Systems Software is a vast improvement over the original 8K ATARI BASIC cartridge. However, it suffers from one major drawback in that it is not readily compatible with the BIT3 80 column board. The people at O.S.S. are not as shortsighted as it would seem for them to leave out 80 column compatibility when they designed the new BASIC. The problem lies with the way the bank selects are implemented in the 800 motherboard. Both the BASIC XL cartridge and the BIT3 80 column board slots use I/O addresses at $D5XX (D500 thru D5FF) to address hardware latches that switch memory banks in the same address range. The BASIC XL switches ROM banks at address space $A000 through $BFFF and the BIT3 board switches 16 256-byte blocks at $D600 through $D6FF.

The solution to this address contention problem is to physically change the hardware addresses of one of the devices. From a hardware viewpoint, it is probably easier to change the latch addresses for the 80 column board. The hardware I/O addresses are generated on the Operating System board (personality module) by the 74LS138 decoder chip. On that chip, there are presently available two other unused address spaces, D1XX(pin 14) and D7XX(pin 7) that were not brought out onto the motherboard because of pin count limitations on the O.S. board. What could be done is to solder an external jumper wire back to the BIT3 80 column board. On the 80 column board, this wire is to be connected to pins 11 and 12 of the 74LS33 NOR gate chip. These two I.C. pins would have to be isolated from the board by lifting them out of the socket and soldering the jumper wire from the personality module directly to these pins.

The next step is a little more difficult in that it requires access to EPROM programming equipment. The firmware that installs the 80 column board as the screen device is in a 2732A EPROM located on the BIT3 board. This code must be disassembled and all references to hardware addresses in the D5XX space must be changed to D1XX. Therefore, locations containing $D5 must be changed to $D1. These locations are (hexadecimal addresses):

```
0EE 0FB 171 1EE 1FB 2EE 2FB 3EE 3FB 4EE
4FB 5EE 5FB 6EE 6FB 7FB 8EE 8FB 9B9 9BF
9C3 9C9 9CD 9D3 9D7 9DD 9EE 9FB AEE AFB
BEE BFB CA5 CAB CAF CB5 CB9 CBF CC3 CC9
CCD CD3 CD7 CDD DEE DFB EA5 EAB EAF EB5
EB9 EBF EC3 EC9 ECD ED3 ED7 EDD EEE EFB
FEE FFB

59C 5B8 5BE 5C3 5C9 713 716 74E 754 759
75F 764 767 7A1 7A7 7AC 7B2 7D6 87B 887
88D B04 B0A C0A C0D C3F C45 C4A C50 C55
C58 C93
```

Once this reprogrammed 2732A EPROM is re-installed in the BIT3 board, the switch to the 80 column format from BASIC XL is still performed by executing

```
A=USR(54818)
```

and the return to 40 columns is made by executing

```
A=USR(54879)
```

The USR calls for Microsoft Basic

remain the same.

The machine language routines to perform the 40/80 column switch are different because of the change in the hardware bank select addresses. The correct routine to switch to the 80 column mode is:

```
LDA #$18
STA $D108 ;SELECT BIT3 EPROM BANK 0
JSR $D623 ;INSTALL 80 COL HANDLER
JMP $E474 ;DO SYSTEM RESET
```

The routine to return to 40 columns is

```
LDA #$18
STA $D108 ;SELECT BIT3 BANK 0
JSR $D660 ;REINSTALL SCREEN HANDLER
JMP $E474 ;DO SYSTEM RESET
```

There is bound to be some incompatibility with other programs that are capable of driving the BIT3 80 column board.

This is mainly because of the change in hardware bank select addresses from D5XX to D1XX. A short term solution is to keep available the original version of the BIT3 firmware on a separate 2732A EPROM and install a switch to remove the jumper that was placed between the Operating System board and the BIT3 board. A permanent solution is to use a utility like DISKSCAN by David Young to scan and disassemble the application software for all references to address space D5XX and substitute D1XX. This could be done with a program like LJK's Letter Perfect word processor.
(Retyped by John Scheid)

by
Dick Reaser

Double Density Data Perfect files to Atari Dos Files

At last! At last! It's here! It's here!

You may recall that in my last article I was lamenting the fact that I couldn't convert DOUBLE DENSITY Data Perfect files to ATARI DOS as could be done for SINGLE DENSITY with the programs included with that article.

Well, thanks to that marvel ** The Electronic Bulletin Board System ** , The U.S. Mail, and the unselfish efforts of a couple of Atarian Hackers (not to be confused with Crackers) named Daniel Prince and John Shreffler, there now exists a program that will do just that. I am also indebted to Mike Andrusak and Matt Arrington for their support. Even received help from Norm Draper (Draper Pascal Fame) of Richardson, TX. He wrote the original single density version and had some good ideas for this version.

This really was a group effort and it was very heartening to get the cooperation that I received.

The program is in Basic and is descended from the programs for single density. When used with Smart Dos or MyDos the output files can be in either single or double density. One purpose that this program can be used for is to transmit Data Perfect files over the modem. It also could be used, I'm sure to convert Double Density Data Perfect files to Single Density Data Perfect files. (LJK hasn't come out with a way to do this yet). Next Issue, I'll talk about the "Data Double" which will convert from Single Density Data Perfect files to Double Density Data Perfect files.

The program is available in the Disk Library as DDDP2AT2 and presumes a double drive system. If someone is really interested, it probably wouldn't be too difficult to make a single drive system.

DDDP2AT2 Program (for double drive)

```
1 REM MODIFICATION TO SINGLE DENSITY
ORIGINAL BY NORM DRAPER OF "DAL-ACE".
2 REM MODIFIED BY DICK REASER WITH
TECHNICAL INPUTS FROM DANIEL PRINCE,
JOHN SHREFFLER & NORM DRAPER.
3 REM THANKS ALSO TO SWAMPS BBS (MIKE
ANDRUSAK) AND MATT ARRINGTON FOR THEIR
HELP.
10 REM WILL WRITE TO SINGLE OR DOUBLE
DENSITY DISK. WORKS WITH SMARTDOS OR
MYDOS.
50 DIM T$(256)
51 ?
52 ?
53 ?
54 ? "                    (12/13/84)"
55 ?
56 ? "      CONVERTS DOUBLE DENSITY DATA
"
57                       ?                  "
PERFECT FILES TO ATARI FORMAT"
58 ? "        (REQUIRES 2 DISK DRIVES)"
59 ?
60 ? "INSERT DATA PERFECT DISK IN DRIVE
1":INPUT T$
68 ?
69 ? "INSERT ATARI FORMATTED DISK IN
DRIVE 2 AND ENTER FILESPEC FOR OUTPUT
FILE"
70 ?
71 ? "   SAMPLE = D2:NAME.EXT":INPUT T$
80 OPEN #2,8,0,T$
90 I=1
100 READ A
110 IF A=999 THEN 140
120 POKE 1535+I,A
130 I=I+1:GOTO 100
```

```
140 S=20
150 POKE 769,1:REM SET TO DRIVE 1 INPUT
160 Z=USR(1536,S,ADR(T$))
170 FOR I=0 TO 255
180 A=PEEK(ADR(T$)+I)
190 IF A=0 THEN 270
200 IF A=2 THEN 240
210 IF A=3 THEN 240
220 PUT #2,A
230 ? CHR$(A);
240 NEXT I
250 S=S+1
260 GOTO 150
270 CLOSE #2
280 END
290 DATA 104,104,141,11,3,104,141,10,
        3,104,141,5,3,104,141,4,3,
        169,82,141,2,3,169,64,141,
        3,3,169,0,141,8,3,169
300 DATA 1,141,9,3,32,89,228,96
310 DATA 999
```

That's it!

Dick "I'm Perfect" Reaser

A COMPUTER BENCHMARK

by Ray Maynard
(Reprinted from the June 1984 issue of the Los Angeles ATARI Computer Enthusiasts Newsletter)

In a rare burst of enthusiasm, I recently took it upon myself to compare the execution speeds of the several programming languages I have available to me on my Atari 800. The results may be enlightening to anyone trying to decide on a programming language. I used the "Prime Number Sieve" benchmark printed in the January 1983 issue of BYTE magazine. The program finds the first 1899 prime numbers, i.e., numbers that have no factors other than itself and 1. This program has been run on almost every computer made and seems to have become a "classic" benchmark program.

The program as published in BYTE uses an array of 8191 numbers. Since each element of a numeric array in Atari Basic and all compatible Basics (except Microsoft) take 6 bytes per element, the array alone requires over 49K bytes! And that's without the program! Since the only values stored into the array are 1 and 0, I decided I could do the same thing using a character string DIMensioned 8191 and storing the characters "1" or "0" into it. Since each character in a string takes only one byte, that saved almost 41K bytes, allowing the program to fit into my Atari. I used two methods of storing and retrieving values to/from the string: substring substitution and PEEKing/POKEing directly into the string. An example of substring substitution is:

```
100 DIM ARRAY$(8191)
110 ARRAY$(100,100)="1"
120 ARRAY$(101,101)="0"
```

while an example of the PEEK/POKE technique is:

```
100 DIM ARRAY$(8191)
110 LOCARRAY=ADDR(ARRAY$)
120 POKE LOCARRAY+99,ASC("1")
130 POKE LOCARRAY+109,ASC("0")
```

I have included the timing results from a few other computers to indicate how the Atari matches up. I ran the timings of ACTION!, C65, Atari Basic cartridge and Basic XL; all the other times are shown from magazine articles. The execution times shown are in seconds for 10 iterations of the main loop of the program. You will notice that executing with the screen display disabled increases the speed. This is because the CPU has to spend significant time keeping the screen updated, even when the display isn't being changed.

Notation:

P/P	PEEK/POKE method
str	Substring method
scr on	Screen enabled
scr off	Screen disabled
fast	BASIC XL "fast" mode
slow	BASIC XL not-"fast" mode

Computer & Language	Exec Time
Cray-1	0.110
CYBER 760, Fortran	0.723
VAX 11/780, Berkely C	1.42
Z-80 C	14.
Atari ACTION! scr off	15.1
Atari ACTION! scr on	21.9
IBM PC, C	22
IBM PC, FORTH	70
BASIC XL fast P/P scr off	101
BASIC XL slow P/P scr off	135.7
BASIC XL fast P/P scr on	148.6
Apple][, Apple Pascal	160
Apple][, FORTH	190
BASIC XL slow P/P scr on	196.5
TRS-80 Mod II, UCSD Pascal	274
Atari, C65	285.7
BASIC XL fast str scr off	1261
TRS-80 Mod II, Basic	1430
BASIC XL slow str scr off	1583
BASIC XL fast str scr on	1847
Apple][, Integer Basic	1850
IBM PC, Basic (integer)	1950
BASIC XL slow str scr on	2299
Atari, Basic P/P scr off	2377
IBM PC, BASICA (flot. pt.)	2400
Atari, Basic str scr off	2416
Apple][, Applesoft Basic	2806
TRS-80 Mod III, Basic	2880
Atari, Basic P/P scr on	3492
Atari, Basic str scr on	3543
TI 99/4A, TI-Basic	3960

I have included below listings of some of the programs that I ran so that you can compare the coding techniques used in the various languages. In the BASIC programs, a loop is used to set all elements of the array to "true" or 1. It would have been much faster to use the code sequence

```
FLAGS$="1":FLAGS$(8191)="1":FLAGS$(2)=F
LAGS$
```

but I didn't in order to stay as close as possible to the original program as presented in BYTE.

I had lots of fun (and frustration) running these benchmarks.

I hope the results are of interest to you.

```
0 REM ATARI BASIC VERSION
10 DIM FLAGS$(8191)
11 DIM FALSE$(1),TRUE$(1)
12 FALSE$="0"
13 TRUE$="1"
14 OPEN #1,4,0,"K:"
15 MAXITER=1:REM MAXIMUM ITERATIONS
20 PRINT MAXITER;" iterations"
21 PRINT "Press any key to start."
22 GET #1,KEY
23  REM SV559=PEEK(559):POKE(559,0):REM
REMOVE "REM" TO DISABLE SCREEN
30   FOR  M=1   TO   MAXITER:REM  START
ITERATIONS LOOP
40 COUNT=0
50 FOR I=1 TO 8191
60 FLAGS$(I,I)=TRUE$
70 NEXT I
80 FOR I=1 TO 8191
90 IF FLAGS$(I,I)=FALSE$ THEN 170
100 PRIME=I+I+3
105 REM PRINT "PRIME=";PRIME
110 K=I+PRIME
120 IF K>8191 THEN 160
130 FLAGS$(K,K)=FALSE$
140 K=K+PRIME
150 GOTO 120
160 COUNT=COUNT+1
170 NEXT I
180 NEXT M
181 REM POKE 559,SV559:REM REMOVE "REM"
TO DISABLE SCREEN
```

Note how the indented listing produced by BASIC XL clearly shows the structure of the FOR loops and the ranges of the IF statements. Also note the use of the WHILE statement:

```
0 Rem  BASIC  XL  VERSION  WITH SCREEN
ENABLED.
1 Rem TO DISABLE THE SCREEN, REMOVE THE
"REMS" IN 22 AND 181
2 Fast
10 Dim Flags$(8191)
11 Dim False$(1),True$(1)
12 False$="0"
13 True$="1"
14 Open #1,4,0,"K:"
20  Print "2  iterations.":Print "Press
any key to start."
21 Get #1,Key
22 Rem SV559=PEEK(559):POKE 559,0
30    For   M=1  To   2:Rem  NUMBER  OF
ITERATIONS
40   Count=0
50     For I=1 To 8191
60     Flags$(I,I)=True$
70     Next I
80     For I=1 To 8191
90     If Flags$(I,I)=False$ Then 170
100    Prime=I+I+3
105    Rem Print " Prime=";Prime
110    K=I+Prime
120      While K<=8191
130      Flags$(K,K)=False$
140      K=K+Prime
150      Endwhile
160    Count=Count+1
170    Next I
180  Next M
181 Rem POKE 559,SV559
190 Print Count;" primes.";CHR$(253)
200 End
```

```
0050 /*  Optimized Systems Software C65
compiler version */
0100 #DEFINE TRUE 1
0200 #DEFINE FALSE 0
0300 #DEFINE SIZE 8190
0301 #DEFINE SIZEP1 8191 /* SIZE PLUS 2
*/
0400   CHAR FLAGS(SIZEP1);
0500 MAIN() $(
0600   INT I,PRIME,K,COUNT,ITER;
0700   PUTS("10 iterations.");
0800   ITER = 1;
0900   WHILE (ITER <= 10 ) $(
1000     COUNT = 0;
1100     I = 0;
1200     WHILE ( I <= SIZE ) $(
1300       FLAGS(I) = TRUE;
1400       I++;
1500     $)
1600   I = 0;
1700     WHILE (I <= SIZE ) $(
1800       IF FLAGS(I) $(
1900         PRIME = I + I +3;
2000         K = I + PRIME;
2100         WHILE ( K <= SIZE) $(
2200           FLAGS(K) = FALSE;
2300           K = K + PRIME;
2400         $)
2500         COUNT++;
2600       $)
2700       I++;
2800     $)
2900     ITER++1;
3000   $)
3050   PUTS("DONE.");
3100 $)
9990 #ASM D1:IO.LIB
```

```
; Optimized  System  Software  ACTION!
compiler version.
DEFINE true="1"
DEFINE false="0"
DEFINE size="8190"
CHAR ARRAY flags(size+1)
CARD i, prime, k, count, iter
BYTE tick=20,
     tock=19


PROC Sieve()
  tick = 0
  tock = 0
  PrintF("10 iterations%E")
  FOR iter = 1 TO 10
  DO
    count = 0
    FOR i = 0 TO size
    DO
      flags(i) = true
    OD
    FOR i = 0 TO size
    DO
      IF flags(i) THEN
        prime = i+i+3
        ; PrintCE(prime)
        k = prime + 1
        WHILE k <= size
        DO
          flags(k) = false
          k ==+ prime
        OD
        count ==+ 1
      FI
    OD
  OD
  i = tick+256*tock
  PrintF("%U  primes   done   in   %U
ticks.%C%E",
    count, i, 253)
  RETURN
```

OUR FIRST SURVEY

by
Terry Craig

Please remove the center sheet from the newsletter and fill it out. The results of this questionaire can be a valuable survey on the reliability of the various disk drives on the market (and serve as a useful buyer's guide), but it requires a substantial number of respondents to be valid and useful.

This sheet has my address on the back and can be folded in thirds and mailed to me or can be collected at the next meeting. If we get a substantial response to this survey we will do the same for printers and modems. The results of this survey will be printed in the next newsletter.

A few minutes of your time now will provide dividends to us all in the future. Thanks for your cooperation!

ATR-8000 Special Interest Group

by Dick Reaser,
Secretary

The next meeting will be held on Tuesday, January 15th at John's house promptly at 7:15 PM. There will be a demonstration of the use of Word Star for us. Additional information about the SIG may be obtained from Hans Karge, our chairman, at (213) 826-8778.

****

Group Addresses

****

Newsletter articles and ads:
Bill Skurski
10393 Almayo Ave.
L.A., CA 90064

Newsletter Exchange:
Dick Reaser
5611 Sherbourne Dr.
L.A., CA 90056

All other information including membership dues:
WLAAUG
P.O. Box 84-396
L.A., CA 90073

BOLT'S ELECTRONIC REPAIR SERVICE
15737 Lemarsh Street Sepulveda, CA 91343
818/893-9664 (Evenings)

ATARI COMPUTER AND DISK DRIVE SERVICE
AND
TANDON 5¼-8" DRIVES AND VIDEO MONITORS

*The most experienced and best equipped service group in Southern California for the service and repair of **ATARI COMPUTER SYSTEMS AND DISK DRIVES.** Each technician has been factory trained and has at least three years experience in an **ATARI** Factory Authorized Service Center.*

Typical/Average Service Charges, Including Parts and Labor(*):

400/800 Computers	*$32.00*
810/1050 Disk Drives	*$36.00*
Complete Computer Diagnostics Checkout and Tune Up (Reseat all I.C.'s, Clean Connector Contacts)	*$16.00*
Complete Disk Drive Checkout (810 and 1050), Tune Up and Lube	*$18.00*

(*) The above typical charges are averages based on service records of over 1,200 disk drives and 900 computers. These typical charges do not include replacement of major assemblies such as complete circuit boards, keyboards, motors or head assemblies.

WEST L.A. ATARI USERS GROUP QUESTIONAIRE #1

DISK DRIVES

REF* #	MAKE	MODEL	SD/ DD?	PURCHASE DATE (M/Y)	VENDOR	PRICE
1						
2						
3						
4						

*Use this reference number in all questions--i.e. the drive referred to in line 1 of this question should be the same as the one referred to in line 1 of all other questions.

ON A SCALE OF 1 TO 10 (10 BEING BEST) HOW DO YOU RATE YOUR DRIVES?

REF #	RELIABILITY	QUALITY OF CONSTRUCTION	FEATURES	DOCUMENTATION	MANUFACTURER SUPPORT	VALUE FOR PRICE
1.						
2.						
3.						
4.						

HAVE ANY DRIVES NEEDED REPAIR?

DRIVE REF #	FAULT	REPAIR FACILITY	QUALITY OF WORK (1 - 10)	FAIR PRICE (Y/N)	DO YOU RECOMMEND (Y/N)

IF YOU HAD TO REPLACE YOUR DRIVES WHAT WOULD YOU BUY?

REF* #	I WOULD REPLACE THIS DRIVE WITH	REASON
1.		
2.		
3.		
4.		

*Reference number refers to drives listed at top of sheet.

WHAT IS YOUR FAVORITE DOS? ____________________

WHY ____________________

TERRY CRAIG
719 SUPERBA AVE
VENICE, CA. 90291

ELEGY

ELEGY WRITTEN IN A COUNTRY CHURCHYARD ON AN ATARI COMPUTER

By Thomas Gray

(Reprinted from the Jersey Atari Computer Group Newsletter, April 1984)

The curfew tolls the knell of parting day,
The lowing herds wind slowly o'er the lea,
His homeward way the ploughman, weary, plods
His homeward way the weary ploughman plods
His homeward weary way the ploughman plods
His way the ploughman homeward, weary, plods
His way the ploughman, weary, homeward plods
His way the weary ploughman homeward plods
His weary way homeward the ploughman plods
His weary way the ploughman homeward plods
Homeward his way the ploughman, weary, plods
Homeward his way the weary ploughman plods
Homeward his weary way the ploughman plods
Homeward the ploughman plods his weary way
Homeward the weary ploughman plods his way
The ploughman, homeward, weary, plods his way
The ploughman homeward plods his weary way
The ploughman plods his weary homeward way
The ploughman, weary, his homeward way plods
The ploughman, weary, homeward plods his way
The ploughman, weary, plods his homeward way
The weary ploughman plods his homeward way
The weary ploughman homeward plods his way
Weary, his homeward way the ploughman plods
Weary, the ploughman his homeward way plods
Weary, the ploughman homeward plods his way
Weary, the ploughman plods his homeward way
And leaves the world to darkness and to me.

(Retyped by John Scheid)

COMPUTER PRINTER STOP 'N SHOP.

We've got more than 20 brands—over 50 different printers! Plus the expertise to help you make the right choice!

NEW HR 35 IN STOCK!

NEW COLOR PRINTER IN STOCK!

Brother HR-15, 25 & 35. Cost effective letter quality. Word processing skills. Opt. tractor feed, sheet feeder, keyboard. 12.5, 23 & 35 cps.

C. Itoh. F10 and A10 letter quality printers, from 18 to 55 cps. Dot matrix printers: models 1550, 8510 and 8600, from 120 to 180 cps. Some with color.

Dataproducts. Dot, daisywheel and line printers: Dataproducts B-300/600, M and DP Series; plus Dataproducts/IDS Prism and Dataproducts/IDS 480.

Epson RX, FX Series. 80 to 160 cps industry standard dot matrix printers. Dot addressable graphics. Narrow and wide carriage.

Infoscribe. 500, 1000, 1100 and 1500 dot matrixes. Heavy duty, from 150 to 400 cps. Graphics. Dual and triple print modes.

Mannesman Tally. Spirit, Pixie Plotter, plus models 160L, 180L and 1600/1800 for a full range of needs. From 80 to 200 cps.

NEC. 2000, 3500 and 7700 Series. 20 to 55 cps with word processing skills. Unique, 128-character print thimble.

Okidata—Full Line. Microline and Pacemark Series dot matrixes. 80 to 350 cps. Single, dual and triple modes. Graphics.

Qume. The affordable 20 cps LetterPro 20. Plus Sprint Series 11/40 and 55 heavy duty word processing printers, with 40 and 55 cps respectively.

- ★ Plus Anadex, Qantex, Siemens, Teletex and Toshiba!
- ★ Demonstrations with 5 popular microcomputers.
- ★ Largest in-stock selection.
- ★ Complete set-up and software support.
- ★ All sorts of accessories.
- ★ Service center for post-sale help.

The "We're here to help" computer printer store™

1703 Stewart Street
Santa Monica, CA 90404
Telephone (213) 828-6436
Mon.- Fri. 10 to 7.
Wed. 10 to 10.
Sat. 10 to 6.
Closed Sunday.

FREE! Come in and ask for our helpful, 24-page printer guide free!

West Los Angeles Atari Users Group
P. O. Box 84-396
Los Angeles, CA 90073

West Los Angeles Atari Users Group
